Neurocomputing 102 (2013) 111–124

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Feature selection for nonlinear models with extreme learning machines

Frénay Benoît [a,b,*], Mark van Heeswijk [b], Yoan Miche [b], Michel Verleysen [a], Amaury Lendasse [b,c,d]

[a] Machine Learning Group, ICTEAM Institute, Université catholique de Louvain, BE 1348 Louvain-la-Neuve, Belgium
[b] Aalto University School of Science, Department of Information and Computer Science, P.O. Box 15400, FI-00076 Aalto, Finland
[c] IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain
[d] Computational Intelligence Group, Computer Science Faculty, University Of The Basque Country, Paseo Manuel Lardizabal 1, Donostia/San Sebastián, Spain

ARTICLE INFO

Available online 7 June 2012

Keywords:
Extreme learning machines
Regression
Feature selection
Regularisation

ABSTRACT

In the context of feature selection, there is a trade-off between the number of selected features and the generalisation error. Two plots may help to summarise feature selection: the feature selection path and the sparsity-error trade-off curve. The feature selection path shows the best feature subset for each subset size, whereas the sparsity-error trade-off curve shows the corresponding generalisation errors. These graphical tools may help experts to choose suitable feature subsets and extract useful domain knowledge. In order to obtain these tools, extreme learning machines are used here, since they are fast to train and an estimate of their generalisation error can easily be obtained using the PRESS statistics. An algorithm is introduced, which adds an additional layer to standard extreme learning machines in order to optimise the subset of selected features. Experimental results illustrate the quality of the presented method.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Feature selection is an important issue in machine learning. On the one hand, if not enough features are selected, prediction may be impossible. On the other hand, using all features may reveal impossible since the amount of available training data is usually small with respect to dimensionality. Aside from generalisation concerns, feature selection may also help experts to understand which features are relevant in a particular application. For example, in cancer diagnosis, feature selection may help to understand which genes are oncogenic. In industry, it is interesting to know which measures are actually useful to assess the quality of a product, since it allows reducing the measurement costs.

Usually there exists a trade-off between the number of selected features and the generalisation error [1]. Indeed, more features means more information, so an ideal model should perform better. However, the curse of dimensionality and the finite number of samples available for learning may harm this ideal view when too many features are considered. Another issue is that the best generalisation error is often not the only objective; interpretability of the selected features may also be a major requirement. Therefore there is often a need for the user to select the number of features by hand, with the help of appropriate tools.

For each fixed number of selected features, one may find (at least in principle) the optimal subset of features, giving the best generalisation error. However choosing between the subsets created in this way for various sizes might be difficult. Two plots may help to summarise feature selection: the feature selection path and the sparsity-error trade-off curve. The feature selection path shows the best feature subset for each subset size, whereas the sparsity-error trade-off curve shows the corresponding generalisation errors. From these plots, experts can choose suitable feature subsets and extract useful domain knowledge. Notice that the feature selection path and the sparsity-error trade-off curve are strongly related, for the latter allows choosing a feature subset in the former.

In real learning situations, the feature selection path and the sparsity-error trade-off curve can only be estimated, since both the target function and the data distribution are unknown. For linear regression problems, the LARS algorithm [2] is an efficient tool for finding the best features for linear models. However, the problem remains open for nonlinear regression problems and models.

For nonlinear problems, ranking methods can be used to rank features using e.g. mutual information [3,4]. Thereafter, feature subsets are built by adding features in the order defined by the ranking. However, feature subsets can evolve discontinuously for nonlinear problems: the best feature subset of size $d+1$ does not necessarily contain the best subset of size d [1]. Methods like

* Corresponding author at: Machine Learning Group, ICTEAM Institute, Université catholique de Louvain, BE 1348 Louvain-la-Neuve, Belgium. Tel.: +321 04 78 133; fax: +321 04 72 598.
E-mail address: benoit.frenay@uclouvain.be (F. Benoît).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2011.12.055

forward or backward search [1] allow searching through the space of possible feature subsets, but they can only select or drop one feature at a time. Moreover, many possible feature selections must be considered at each iteration by such methods based on greedy search.

This paper proposes a new algorithm to build the feature selection path and the sparsity-error trade-off curve for nonlinear problems. Contrarily to e.g. forward search, the proposed iterative algorithm considers only one neighbour at each iteration. Yet, multiple features can enter or leave the current feature subset at each step of the search. Extreme learning machines are used since they are very fast to train and an estimate of their generalisation error can easily be computed [5–10]. The proposed method is theoretically and experimentally compared with other feature selection methods. Experiments show that the proposed algorithm obtains reliable estimates of the two plots: the feature selection path and the sparsity-error trade-off curve. In some cases, the proposed algorithm obtains (i) optimal test errors using less features and (ii) feature selection paths with more information, with respect to the paths obtained by the other feature selection algorithms used here for comparison.

The following of this paper is organised as follows. Section 2 discusses feature selection. Section 3 introduces the feature selection path and the sparsity-error trade-off curve and discusses how they can be used in practice. Section 4 proposes an algorithm and compares it theoretically with existing methods. Section 5 assesses the proposed algorithm experimentally and conclusions are drawn in Section 6.

2. Domain analysis and feature selection

In many applications, feature selection is necessary. Indeed, the number of available samples is usually small with respect to the data dimensionality. In that case, the curse of dimensionality prevents us from using all the features, since the necessary number of training samples grows exponentially with the dimensionality. Therefore, feature selection consists of choosing a trade-off between the number of selected features and the adequacy of the learned model. However, it is not always obvious what is a *good* feature subset.

A common criterion for assessing the quality of a subset of features is the generalisation error, i.e. the expected error for new samples. This criterion relates to the capacity of the model to generalise beyond training data. Sometimes, experts simply want to minimise the generalisation error. However, in some contexts, experts are searching for sparse feature subsets with only a few features because interpretability is a major concern. In such cases, the number of features is chosen in order to achieve sufficient generalisation. Limiting the number of features may also be necessary because of e.g. measurement costs. In conclusion, feature selection requires flexible tools which are able to adapt to specific user needs.

The next section discusses two strongly related tools for addressing common questions in feature selection situations: the feature selection path and the sparsity-error trade-off curve. Section 4 proposes an algorithm to estimate both of them in the case of nonlinear regression problems. In this paper, the focus is set on regression and the mean square error (MSE)

$$\frac{1}{n}\sum_{i=1}^{n}[t_i-\hat{f}(x_i^1,..,x_i^d\,|\,\theta)]^2 \tag{1}$$

is used, where $x_i=(x_i^1,\ldots,x_i^d)$ is instance i, t_i is the target value, n is the number of samples and $\hat{f}$ is a function approximator with parameters θ.

3. Feature selection path and sparsity-error trade-off curve

Given a set of features, a feature selection path (FSP) shows the best feature subset for each subset size. Here, best feature subsets are selected in terms of generalisation error. Fig. 1 shows an estimate of the feature selection path (FSP) for an artificial problem, called here the XOR-like problem. The artificial dataset is built using six random features which are uniformly distributed in [0,1]. For each sample $x_i=(x_i^1,\ldots,x_i^6)$, the target is

$$f(x_i)=x_i^1+(x_i^2>0.5)(x_i^3>0.5)+\epsilon_i \tag{2}$$

where (i) $(x>0.5)$ is equal to 1 when $x>0.5$ and is equal to 0 otherwise and (ii) ϵ_i is a noise with distribution $\mathcal{N}(0,0.1)$. This regression problem is similar to the XOR problem in classification: the product term can only be computed using both features 2 and 3. In order to have a sufficient number of data for the feature selection, 1000 training samples were generated. Fig. 1 is obtained with the approach proposed in this paper (see Sections 4 and 5). Each column corresponds to a subset size, where black cells correspond to selected features. Rows correspond to features. In essence, a feature selection path is very similar to the plots in Efron et al. [2], which show estimates of regression coefficients for different coefficient sparsities.

Each feature subset corresponds to a generalisation error. Indeed, for each subset size, one can estimate how well the selected features allow generalising to new samples. These generalisation errors are required in order to choose one of the feature subsets in the FSP. Therefore, one obtains a sparsity-error trade-off (SET) curve, which shows the best achievable generalisation error for the different feature set sizes. Here, sparsity refers to the size of the feature subset itself: sparse feature subsets contain less features.

Fig. 2 shows an estimate of the sparsity-error trade-off (SET) curve for the XOR-like problem, where the generalisation errors correspond to the feature subsets given in Fig. 1. The SET curve shows that the generalisation error is large when only a few features are selected, i.e. when the feature subset is too sparse. The generalisation error improves quickly as sparsity decreases and achieves its optimum for three features. Then, the generalisation error starts to increase, because of the curse of dimensionality. Indeed, the number of training samples becomes too small with respect to the dimensionality.

Using the feature selection path and the sparsity-error trade-off curve, experts can answer many questions. It is possible to see e.g. which features are useful, which features are necessary to achieve correct results, which features do not seem to be worth collecting, etc. These questions cannot be answered if one only

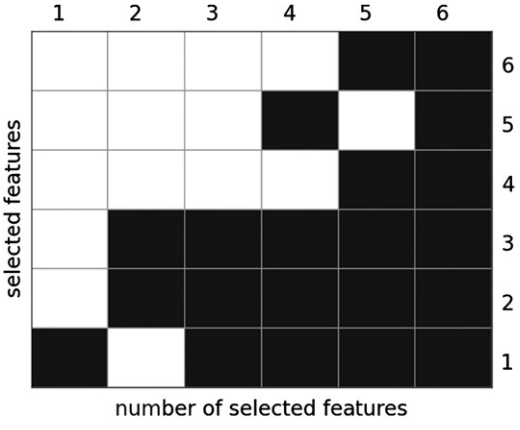

Fig. 1. Estimate of the feature selection path for the XOR-like problem. Columns and rows correspond to subset sizes and features, respectively.

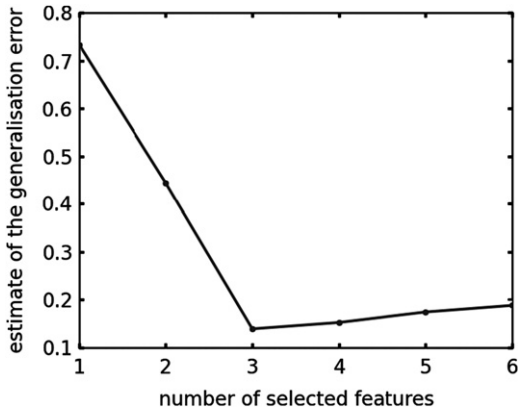

Fig. 2. Estimate of the sparsity-error trade-off curve for the XOR-like problem.

has the best feature subset: the path of feature subsets is necessary, as well as the corresponding generalisation errors.

Let us shortly discuss the XOR-like problem using the FSP and the SET curve in Figs. 1 and 2. Here, three features are sufficient to achieve optimal models. Indeed, the estimate of the generalisation error has reached its minimum value. Notice that the selected features are the relevant features in Eq. (2).

The FSP provides important additional information: features 2 and 3 should be selected together. Indeed, when only one feature is selected, the feature subset is {1}. But when two feature are selected, feature 1 is no longer used. Instead, features 2 and 3 are selected jointly. This cannot be seen when looking only at the optimal feature subset { 1,2,3}. The FSP reflects Eq. (2), where the target depends on a nonlinear combination of features 2 and 3.

4. Estimating FSPs and SET curves

In practice, the true FSP and the true SET curve are impossible to obtain. Indeed, both the true approximated functional and the true data distribution are unknown. Instead, one has to rely on estimates. This section reviews existing approaches and introduces a new algorithm in order to overcome their weaknesses.

4.1. Estimating the generalisation error

In order to estimate the SET curve, it is necessary to choose an estimator of the generalisation error. The generalisation error corresponds to the expected value of the error on new, unknown samples. Hence, techniques like e.g. cross-validation or bootstrap can be used [11,12]. Namely, these methods use the available data to build a training set and a test set. A model is trained using training data and tested on test data. The resulting error gives an estimate of the generalisation error, since none of the test samples have been used for training. The process can be repeated to obtain reliable estimates.

It should be pointed out that both cross-validation and bootstrap estimate the generalisation of a given model, not the best possible generalisation error. Therefore, using a good model is necessary to obtain a reliable estimate of the SET curve. A problem might be that the choice of the feature subsets may be biased by the model. Indeed, it is possible for optimal feature subsets to differ with respect to the model. However, it seems reasonable to think that the problem will not be too important for sparse feature subsets, which are precisely the feature subsets which are looked for by experts.

In this paper, leave-one-out (LOO) cross-validation [13] is used to estimate the generalisation error. First, a single sample is removed from the dataset and a model is built using the remaining data. Then, the prediction error on the unused sample is computed. The process is repeated for each sample; the average result gives an estimate of the generalisation error.

4.2. Optimising feature subsets

In practice, it is impossible to test all possible feature subsets, since the number of tests grows exponentially with the dimensionality of data. Instead, one typically starts with an arbitrary feature subset, which is iteratively improved. Examples of such methods include LARS and forward-backward search. The latter can e.g. use mutual information to guide the search.

LARS [2] is an algorithm which solves efficiently the LASSO problem [14], i.e. an L_1-regularised linear regression. The constraint on the L_1-norm enforces sparsity: the number of selected features increases as the regularisation decreases. LARS can be used for feature selection and the path of its solutions can be converted into a FSP. However, LARS is optimal for linear problems but not necessarily for nonlinear ones.

Mutual information [3,4] is a measure of the statistical dependency between a set of features and a target variable. It can be used to choose a subset of features using the strength of the statistical link between the subset and the output. A simple example of feature selection method based on mutual information consists of (i) ranking features according to their mutual information with respect to the output and (ii) adding features to the feature subset in the order defined by the ranking. In such a case, only d features subsets need to be considered, where d is the dimensionality. Such procedures are simple, but they cannot deal efficiently e.g. with XOR-like problems, where features must be considered together to establish statistical dependencies. An alternative consists of using multivariate greedy methods, like e.g. forward or backward search.

Forward search [1] starts from an empty set of features and iteratively selects a feature to add. Backward search [1] is similar, but its starts with all features and iteratively removes them. At each step, every feature which is not yet selected has to be considered, which means that a total of $\mathcal{O}(d^2)$ feature subsets are considered. Mutual information or validation error can be e.g. used to choose feature subsets and guide the search. Since features are added (or removed) one at a time, successive feature subsets can only differ by one feature, which may not be optimal in practice.

In the above methods, it is impossible to add or remove several features simultaneously. It means that for problems like the XOR-like problem of Section 3, the FSP may not be optimal and may not highlight the fact that some features must be selected together. Indeed, Fig. 1 shows that when the number of selected features changes from one to two, three features must be changed. This cannot be achieved with e.g. forward search. Moreover, for the above methods, a lot of possible feature subsets have to be considered at each iteration.

In the rest of this section, a new algorithm is introduced to overcome the weaknesses of the above methods. Namely, the proposed algorithm allows obtaining FSP with significant differences in successive feature subsets. In Section 5, experiments show that in some situations, the proposed algorithm obtains (i) optimal test errors using less features and (ii) FSPs with more information than the FSPs obtained by LARS and two other greedy search algorithms.

4.3. Relaxing the feature selection problem

The generalisation error is seldom used to guide the search for feature subsets. Indeed, this error is usually very costly to

estimate, since one needs to rely on e.g. cross-validation. Instead, the heuristic methods described above use other objective functions like e.g. regularised training error or mutual information. Here, a similar approach to LARS is proposed. The feature selection problem is firstly relaxed and a regularisation scheme is used to enforce feature sparsity.

In order to approximate the FSP and the SET curve, let us focus on finding good feature subsets and good models for each feature subset size. Using Eq. (1), the corresponding problem can be stated for regression as

$$\min_{\beta,\theta} \frac{1}{n}\sum_{i=1}^{n}[t_i - \hat{f}(\beta_1 x_i^1,..,\beta_d x_i^d|\theta)]^2 \quad \text{s.t. } \|\beta\|_0 = d_s \le d \tag{3}$$

where β is a vector of binary variables s.t. $\beta_i \in \{0,1\}$, $\|\beta\|_0$ is the L_0-norm of β, i.e. the number of non-zero components β_i, and d_s is the size of the feature subset. Here, each binary variable β_i indicates whether the ith feature is selected or not. The constraint limits the number of active features. Notice that the generalisation error is replaced by the training error in (3).

Because of the L_0-norm constraint, the above optimisation problem is still combinatorial and difficult to solve. In order to simplify the optimisation problem, let us first rewrite Eq. (3) as a regularisation, i.e.

$$\min_{\beta,\theta} \frac{1}{n}\sum_{i=1}^{n}[t_i - \hat{f}(\beta_1 x_i^1,..,\beta_d x_i^d|\theta)]^2 + C_0\|\beta\|_0 \tag{4}$$

for some regularisation constant $C_0 \in \Re^+$. It is now possible to use a common approach in machine learning, which consists of replacing the L_0-norm with an L_1-norm [15]. Indeed, it has been shown e.g. for linear models [2] and support vector machines [16,17] that regularising with respect to the L_1-norm decreases the number of features actually used by the model. Moreover, the L_1-norm is easier to optimise than the L_0-norm. The same idea is used in LARS: [2] shows that a linear regression with an L_1 regularisation can be used to reduce the number of selected features. Notice that the above approach is similar to a common approach in integer programming which is called relaxation [18]. Eq. (4) becomes

$$\min_{\tilde{\beta},\theta} \frac{1}{n}\sum_{i=1}^{n}[t_i - \hat{f}(\tilde{\beta}_1 x_i^1,..,\tilde{\beta}_d x_i^d|\theta)]^2 + C_1\|\tilde{\beta}\|_1 \tag{5}$$

for some regularisation constant $C_1 \in \Re^+$. Vector $\tilde{\beta}$ no longer defines a feature subset. Instead, Eq. (5) is related to *feature scaling*, a problem similar to feature selection where ones tries to find coefficients giving a different importance to each feature.

Eq. (5) is easier to solve than Eq. (4) since it is differentiable. Yet, solutions of Eq. (5) can be converted into approximated solutions of Eq. (4). Indeed, a non-zero $\tilde{\beta}_i$ variable can be considered to mean that the corresponding feature is selected, i.e. $\beta_i = 1$. Indeed, even for small values of $\tilde{\beta}_i$, feature i is still used by the model. The next subsection proposes an algorithm to build the FSP and the SET curve using Eq. (5).

Notice that the C_1 constant is controlling the regularisation on $\tilde{\beta}$. Indeed, the resulting feature scaling becomes sparser and sparser as C_1 increases. In general, an L_1-norm regularisation on a vector of coefficients causes the coefficients to become zero one after another, until none of them remains [14,2,19,1]. Indeed, using the L_1-norm regularisation is equivalent to setting a Laplacian prior on $\tilde{\beta}$ [19]. Using the L_2-norm, sparsity would be lost [19,1], which explains why the L_1-norm is used here. The L_1-norm regularisation behaviour is illustrated by Efron et al. in the case of LARS [2].

4.4. Solving the relaxed feature selection problem

For various values of C_1, the solutions of Eq. (5) have different degrees of sparsity. The algorithm which is proposed here uses this fact to span the different sizes of feature subsets. Indeed, if C_1 is progressively increased, the sparsity of resulting feature subsets will increase as well. In a nutshell, the proposed algorithm therefore simply solves Eq. (5) for increasing C_1 values.

Solving Eq. (5) is not trivial. Indeed, the objective function may be non-convex and many local minima may exist. A possible approach is gradient descent with multiple restarts. However, gradient descent on continuous variables can be very slow, e.g. if the minimised function has many plateaux. Moreover, it is difficult to reach exact values like e.g. $\tilde{\beta}_i = 0$ or $\tilde{\beta}_i = 1$.

In this paper, feature scalings are discretised to overcome the above problems. Indeed, exact solutions are not necessary, since they are converted into binary feature subsets afterwards. The space of all possible feature scaling $[0,1]^d$ becomes a hypergrid $\{0, 1/k,..,1\}^d$ with $k+1$ non-zero values in each dimension. Next, the gradient of the regularised training error is used to guide the search. At each step, the search only considers the direct neighbour pointed to by the gradient. Here, a direct neighbour of the feature scaling $\tilde{\beta}$ is a feature scaling $\tilde{\beta}'$ s.t. $\max_i|\tilde{\beta}_i' - \tilde{\beta}_i| \le 1/k$. According to that definition, several feature scalings can change at each step. In this paper, k is equal to 10 for the experiments.

The proposed procedure is detailed in Algorithm 1. A fast implementation based on extreme learning machines is proposed in Section 4.5. For each repetition of the main loop, the feature scaling $\tilde{\beta}$ is randomly initialised and C_1 is set to zero, i.e. no regularisation is initially performed. The current solution and the current model are used to update the FSP and the SET curve for $\|\tilde{\beta}\|_0$ features, if necessary. Given the current solution $\tilde{\beta}$ and the current value of C_1, the gradient of the regularised training error is used to find a candidate $\tilde{\beta}_{new}$ in the direct neighbourhood of $\tilde{\beta}$. If $\tilde{\beta}_{new}$ is actually better than $\tilde{\beta}$ in terms of regularised training error, then $\tilde{\beta}_{new}$ becomes the new, current solution. Otherwise, a local minimum has been found; C_1 is increased and the algorithm searches for a sparser solution with a smaller regularised training error (with respect to the new C_1 constant). The algorithm stops when C_1 is so large that the L_0-norm $\|\tilde{\beta}\|_0$ becomes zero, i.e. when the feature subset becomes empty.

Algorithm 1. Local search algorithm for the relaxed feature selection problem

for all restarts **do**
  $C_1 \leftarrow 0$
  initialise $\tilde{\beta}$ randomly
  find the vector of parameters θ corresponding to $\tilde{\beta}$ (train a model)
  compute the regularised training error

  while $\|\tilde{\beta}\|_0 > 0$ **do**
    estimate the generalisation error obtained using $\tilde{\beta}$ and θ
    convert the feature scaling $\tilde{\beta}$ into a feature subset β
    update the FSP and the SET curve, if necessary

    compute the gradient of the regularised training error
    find the direct neighbour $\tilde{\beta}_{new}$ pointed by the gradient
    find the vector of parameters θ_{new} corresponding to $\tilde{\beta}_{new}$ (train a model)

    compute the new regularised training error
    if the regularised error has not decreased **then**
      increase C_1 until the gradient points to $\tilde{\beta}_{new}$ s.t. $\|\tilde{\beta}_{new}\|_1 < \|\tilde{\beta}\|_1$

increase C_1 until the regularised training error for $\tilde{\beta}_{new}$ becomes lower

find the vector of parameters θ_{new} corresponding to $\tilde{\beta}_{new}$
compute the new regularised training error
end if

update the current solution $\tilde{\beta}$ with $\tilde{\beta}_{new}$
update the vector of parameters θ with θ_{new}
update the regularised training error
end while
end for

In Algorithm 1, θ is the vector of model parameters introduced in Eq. (1). The procedure to obtain θ depends on the type of model which is used. For example, in the case of linear regression, instances can be first multiplied by scaling coefficients $\tilde{\beta}$. Then, the weights θ of the linear regression are obtained as usual using the scaled instances and the target values. The case of non-linear models is illustrated in Section 4.5, which proposes a fast implementation of Algorithm 1 based on extreme learning machines.

Since (i) each local minimum is reached in a finite number of steps and (ii) C_1 is increased whenever a local minimum of the regularised training error is reached, Algorithm 1 is guaranteed to terminate in a finite amount of steps. Eventually, the feature subset becomes empty and the algorithm terminates. Feature scalings are converted into feature subsets by simply assuming that features with non-zero scalings $\tilde{\beta}_i$ are selected. Indeed, simply rounding the scalings toward 0 or 1 could not be sufficient, as even features which correspond to small scalings may nevertheless be used by the model.

Algorithm 1 is not guaranteed to find the optimal solution for each feature subset size. However, by slowly increasing the regularisation on $\|\beta\|_1$, the proposed algorithm spans the whole spectrum of feature subsets sizes. Multiple restarts are performed to decrease the influence of local minima.

Compared with e.g. forward search and backward elimination, Algorithm 1 has several advantages. Firstly, the gradient information is used to consider only one neighbour at each iteration. Secondly, multiple features can be updated simultaneously. Moreover, Algorithm 1 can select unselected features or remove selected features, which is impossible in simple forward or backward search.

4.5. Fast implementation of the proposed algorithm

Algorithm 1 requires (i) models which are fast to train and (ii) a fast estimator of the generalisation error. Extreme learning machines (ELMs) meet both these requirements [5–8]. Firstly, their training is very fast, since it only requires solving a linear system. Secondly, the LOO error of an ELM can be computed quickly and exactly using the PRESS statistics [13,10]. The LOO error is a special case of the cross-validation error, an estimator of the generalisation error. This subsection firstly reviews ELMs, then shows how to use ELMs in order to implement Algorithm 1.

ELMs are feed-forward neural networks with one hidden layer (see Fig. 3). In traditional feed-forward neural networks, the weights of both hidden and output weights are simultaneously optimised through gradient descent. This learning procedure is called back-propagation in the case of the popular multi-layer perceptron [20]. However, gradient descent has many drawbacks. In particular, it is slow and can get stuck in one of the many local minima of the objective function [5].

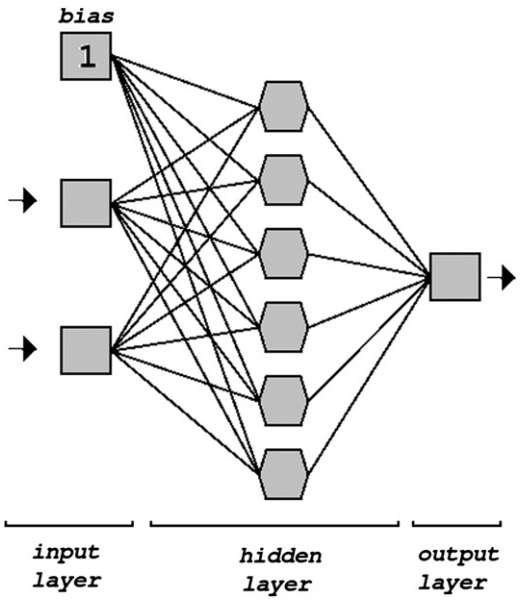

Fig. 3. Feed-forward neural network with one hidden layer.

Extreme learning machines [5–7] provide an interesting alternative to train feed-forward neural networks, which solves the above problems. Firstly, the weights and biases in the hidden layer are set randomly and remain fixed during the training process. Then, the hidden layer output matrix of the ELM with m hidden neurons is computed as

$$H = \begin{bmatrix} \sigma\left(\sum_{i=1}^{d} W_{i1}X_{1i} + b_1\right) & \cdots & \sigma\left(\sum_{i=1}^{d} W_{im}X_{1i} + b_m\right) \\ \vdots & \ddots & \vdots \\ \sigma\left(\sum_{i=1}^{d} W_{i1}X_{ni} + b_1\right) & \cdots & \sigma\left(\sum_{i=1}^{d} W_{im}X_{ni} + b_m\right) \end{bmatrix} \quad (6)$$

where σ is the activation function of the hidden units, W is the $d \times m$ matrix of random hidden layer weights, X is a $n \times d$ matrix where each row corresponds to a training instance and b is the m-dimensional vector of random hidden layer biases. Usually, σ is the hyperbolic tangent tanh, but any infinitely differentiable function can be used [5]. For example, radial basis functions are also considered in [21].

Since the output of an ELM is a linear combination of the m hidden layer neuron outputs, the output weights are found by solving the linear problem

$$\min_w \|T - Hw\|_2^2 \quad (7)$$

where T is an n-dimensional vector containing the target values and w is the m-dimensional vector of output weights. It is well known that the unique solution of Eq. (7) is

$$w = H^{\dagger}T \quad (8)$$

where $H^{\dagger}$ is the Moore–Penrose pseudo-inverse [22] of H. Using e.g. singular value decomposition, $H^{\dagger}$ can be computed efficiently.

In the seminal paper [5], it is shown that ELMs achieve good performances in terms of error, with respect to other state-of-the-art algorithms. Moreover, ELMs are shown to be much faster than traditional machine learning models. For example, they can be trained up to thousands times faster than support vector machines. Notice that there exist a significant number of variants of ELMs. In particular, other activation functions can be used [21] and ELMs can be trained incrementally [9]. The universal approximation capability of ELMs is discussed in [23].

Another advantage of ELMs is that it is possible to obtain an analytical expression for an estimate of their generalisation error

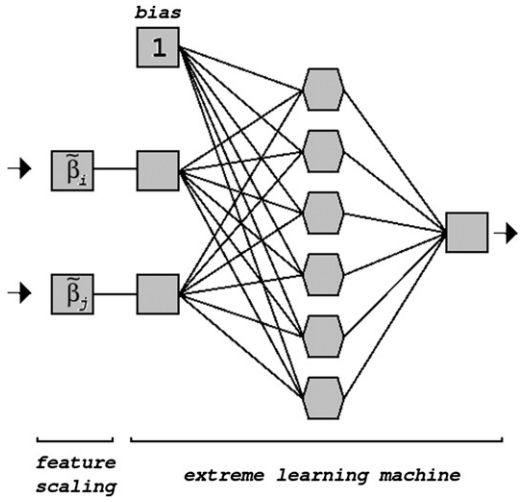

Fig. 4. Extreme learning machine with integrated feature scaling.

[10]. Indeed, the LOO error for an ELM can be obtained using the PRESS statistics [13], i.e.

$$\text{PRESS} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{e_i}{1-z_{ii}}\right]^2 \tag{9}$$

where e_i is the error for the ith training instance and z_{ii} is the ith diagonal term of

$$Z = HH^{\dagger}. \tag{10}$$

Since ELMs are fast to train and a fast estimator of their generalisation error exists, they are perfectly fitted to implement Algorithm 1. Intuitively, as shown in Fig. 4, the feature scaling can be seen as an extra layer put in front of the ELM. In the following, the feature scaling is directly plugged into ELMs to make the development easier. The hidden layer output matrix of the new ELM becomes

$$\tilde{H} = \begin{bmatrix} \sigma\left(\sum_{i=1}^{d} W_{i1}\tilde{\beta}_i X_{1i} + b_1\right) & \cdots & \sigma\left(\sum_{i=1}^{d} W_{im}\tilde{\beta}_i X_{1i} + b_m\right) \\ \vdots & \ddots & \vdots \\ \sigma\left(\sum_{i=1}^{d} W_{i1}\tilde{\beta}_i X_{ni} + b_1\right) & \cdots & \sigma\left(\sum_{i=1}^{d} W_{im}\tilde{\beta}_i X_{ni} + b_m\right) \end{bmatrix} \tag{11}$$

and the optimal output weights of the new ELM are now given by

$$\tilde{w} = \tilde{H}^{\dagger}T \tag{12}$$

Using the above definitions, the gradient of the regularised training error with respect to the scaling vector $\tilde{\beta}$ becomes

$$\nabla_{\tilde{\beta}}\text{MSE} = \begin{bmatrix} -\frac{2}{n}\sum_{i=1}^{n} e_i \sum_{j=1}^{m} w_j W_{1j} X_{i1} \tilde{H}'_{ij} \\ \vdots \\ -\frac{2}{n}\sum_{i=1}^{n} e_i \sum_{j=1}^{m} w_j W_{dj} X_{id} \tilde{H}'_{ij} \end{bmatrix} \tag{13}$$

where e_i is the error for the ith training instance and $\tilde{H}'$ is defined as

$$\tilde{H}' = \begin{bmatrix} \sigma'\left(\sum_{i=1}^{d} W_{i1}\tilde{\beta}_i X_{1i} + b_1\right) & \cdots & \sigma'\left(\sum_{i=1}^{d} W_{im}\tilde{\beta}_i X_{1i} + b_m\right) \\ \vdots & \ddots & \vdots \\ \sigma'\left(\sum_{i=1}^{d} W_{i1}\tilde{\beta}_i X_{ni} + b_1\right) & \cdots & \sigma'\left(\sum_{i=1}^{d} W_{im}\tilde{\beta}_i X_{ni} + b_m\right) \end{bmatrix}$$

$$= \begin{bmatrix} 1-\tilde{H}_{11}^2 & \cdots & 1-\tilde{H}_{1m}^2 \\ \vdots & \ddots & \vdots \\ 1-\tilde{H}_{n1}^2 & \cdots & 1-\tilde{H}_{nm}^2 \end{bmatrix} \tag{14}$$

since σ is here the hyperbolic tangent tanh whose derivative is $\tanh'(z) = 1-\tanh(z)^2$.

Algorithm 1 can be implemented using (i) Eq. (12) to train an ELM, (ii) Eq. (9) to estimate its generalisation error and (iii) Eq. (13) to compute the gradient guiding the search. Notice that the vector of model parameters θ which appears in both Eq. (1) and Algorithm 1 corresponds here to the vector of output weights $\tilde{w}$. In theory, one should optimise the ELM size m before starting the scaling search. However, there is no guarantee that the optimal ELM size is identical for different numbers of selected features. Therefore, the solution chosen here is simply to choose a random ELM size at each restart. Indeed, only ELMs with correct sizes (with respect to the feature subset size) will eventually be taken into account, since they are precisely the ELMs which will be used to build the FSP and the SET curve.

In the rest of this paper, the proposed implementation of Algorithm 1 is called ELM-FS, for ELM-based feature selection.

4.6. Remarks on the estimated SET curve

In the proposed approach, the SET curve is estimated by selecting the best feature subsets among those which are considered during the search. The resulting SET curve can be used to select a feature subset, e.g. the one with the lowest generalisation error. However, two remarks hold here. Firstly, the estimate of the generalisation error provided by cross-validation (and in particular LOO) tends to be less reliable when more and more features are added, because of the curse of dimensionality. This could lead experts to choose too large feature subsets. Secondly, the estimated generalisation error is not valid any more as soon as a particular feature selection is chosen. Indeed, since the estimate was used to select a particular feature subset, it is biased for this particular solution. An additional, independent set of instances should be used to estimate the final generalisation error. Yet, the estimated SET curve can be used to select a subset size.

5. Experiments

In this section, two goals are pursued through experiments. Firstly, it is necessary to assess whether the proposed algorithm obtains feature subsets which are either equivalent or better than those obtained using standard feature selection methods. Secondly, since the proposed algorithm naturally provides a FSP, it is important to assess whether the FSP provides useful information or not, with respect to methods which only provide a best feature subset.

The following of this section is organised as follows. Section 5.1 describes the experimental settings. Sections 5.2 and 5.3 show the results for artificial and real datasets, respectively.

5.1. Experimental settings

ELM-FS is compared with three other methods, in terms of feature subsets and test error: LARS, forward search with mutual information (MI-FW) and forward-backward search with mutual information (MI-FWBW). LARS searches for linear relationships in data [2], whereas MI-FW and MI-FWBW search for more general, possibly nonlinear relationships. Whereas the features and the output are compared in terms of correlation for LARS, mutual information estimates their statistical dependency. Each feature is normalised using the mean and the standard deviation computed on training samples.

MI-FW starts with an empty subset of features. At each iteration, MI-FW computes the mutual information between the current subset of features and the output. Then, the feature which

increases the most this mutual information is added to the current subset of features. The algorithm continues until all features have been added. MI-FW is not repeated, since it always starts with the same, empty subset of features. The implementation of MI-FWBW is similar, except that features can be either added or removed at each step. Moreover, MI-FWBW is repeated 100 times with random initial feature subsets in order to (i) reduce the effect of local minima and (ii) obtain a complete FSP. Mutual information is estimated using a k-nearest neighbours approach introduced by Kraskov et al. [3], where k is chosen using cross-validation [24].

ELM-FS is performed using 100 repetitions. The neurons of the 100 corresponding ELMs are chosen in a fixed set of 100 neurons. For each repetition, (i) a random number of neurons is chosen between 1 and 100 and (ii) the corresponding number of neurons are chosen in the fixed set of neurons.

The test errors are computed as follows. For each dataset, an ELM is initialised with 100 neurons. Then, for each feature selection algorithm and each feature subset size, the output weights are optimised using the feature subset of the corresponding size, the training samples and OP-ELM, a state-of-the-art method in extreme learning [10]. Eventually, the predictions of

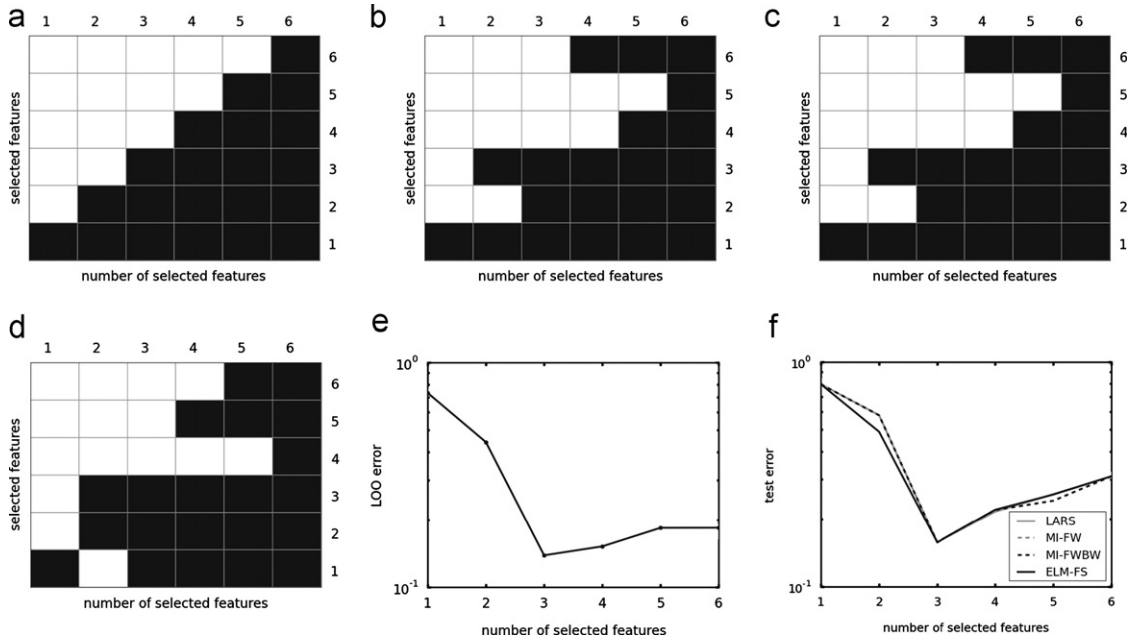

Fig. 5. Results for the XOR-like dataset: (a–d) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (e) the SET curve for ELM-FS and (f) the test errors for the four compared methods. Notice the logarithmic scales for errors. (a) FSP: LARS. (b) FSP: MI-FW. (c) FSP: MI-FWBW. (d) FSP: ELM-FS. (e) SET curve: ELM-FS. (f) test errors.

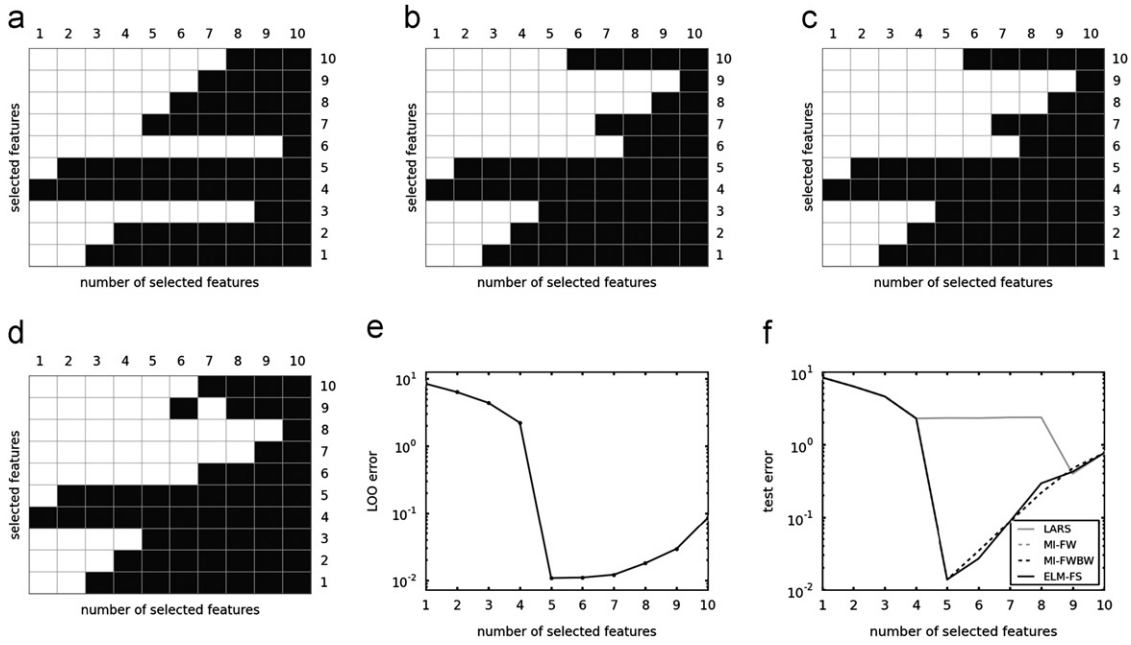

Fig. 6. Results for the functional dataset: (a-d) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (e) the SET curve for ELM-FS and (f) the test errors for the four compared methods. Notice the logarithmic scales for errors. (a) FSP: LARS. (b) FSP: MI-FW. (c) FSP: MI-FWBW. (d) FSP: ELM-FS. (e) SET curve: ELM-FS. (f) test errors.

the resulting ELM are compared on the test samples in order to produce the test error. In order to be able to compare the different feature selection algorithms, the test errors for a given dataset are obtained using the same initial ELM. Therefore, identical feature subsets correspond to identical test errors.

5.2. Results on artificial datasets

In this subsection, two artificial toy problems are used to compare ELM-FS with LARS, MI-FW and MI-FWBW: (i) the XOR-like problem introduced in Section 3 and (ii) a complex, nonlinear functional [24]. For convenience, the definition of the XOR-like problem is repeated below.

For the XOR-like problem, the artificial dataset is built using six random features which are uniformly-distributed in [0,1]. For each sample $x_i = (x_i^1, \ldots, x_i^6)$, the target is

$$f(x_i) = x_i^1 + (x_i^2 > 0.5)(x_i^3 > 0.5) + \epsilon_i \quad (15)$$

where (i) $(x > 0.5)$ is equal to 1 when $x > 0.5$ and is equal to 0 otherwise and (ii) ϵ_i is a noise with distribution $\mathcal{N}(0,0.1)$. This regression problem is similar to the XOR problem in classification: the product term can only be computed using both features 2 and 3.

For the functional problem, the artificial dataset is built using ten random features which are uniformly-distributed in [0,1]. For each sample $x_i = (x_i^1, \ldots, x_i^{10})$, the target is

$$f(x_i) = 10\sin(x_i^1)x_i^2 + 20(x_i^3 - 0.5)^2 + 10x_i^4 + 5x_i^5 + \epsilon_i \quad (16)$$

where ϵ_i is a noise with distribution $\mathcal{N}(0,0.1)$.

Table 1
Computation times in seconds of the different feature selection algorithms for the XOR-like problem and the functional problem, including the search of the k parameter for the Kraskov estimator.

	LARS	MI-FW	MI-FWBW	ELM-FS
XOR-like	1.2e−2	1.0e+2	2.6e+2	3.1e+2
functional	1.3e−2	1.7e+2	5.1e+2	4.2e+2

For both artificial problems, 1000 training samples were generated in order to have a sufficient amount of data for the feature selection. Each test set consists of 9000 samples, so that the test error accurately estimates the generalisation error.

For the XOR-like dataset, Fig. 5 shows (i) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (ii) the SET curve for ELM-FS and (iii) the test errors for the four methods. The SET curve recommends to use three features. In this case, the four methods choose the correct feature subset, i.e. $\{1,2,3\}$. However, the FSP obtained using ELM-FS provides additional information: features 2 and 3 should be selected together. Indeed, when ELM-FS selects only one feature, feature 1 is selected. But when ELM-FS selects two features, feature 1 is no longer used. Instead, features 2 and 3 are selected jointly. This information cannot be seen on the FSPs of LARS, MI-FW and MI-FWBW: they successively select feature 1 and either feature 2 or feature 3. In conclusion, the FSP obtained using ELM-FS reflects well Eq. (15), where the target depends on a nonlinear combination of features 2 and 3. Notice that when only two features are selected, ELM-FS obtains a slightly smaller test error, which supports its choice of features.

For the functional dataset, Fig. 6 shows (i) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (ii) the SET curve for ELM-FS and (iii) the test errors for the four methods. ELM-FS recommends to use the five features which are actually the ones used to compute the target. Identical feature subsets and test errors are obtained using the other algorithms, except LARS which includes feature 3 only for large feature subset sizes and achieves larger test errors.

According to results for the XOR-like and functional datasets, ELM-FS is able to cope with nonlinearities and obtains sound feature subsets. Moreover, for both datasets, the feature subset which corresponds to the minimum of the SET curve also obtains the minimum test error. In other words, the SET curve estimated by ELM-FS using the PRESS statistics is a valuable tool for choosing the size of the optimal feature subset.

An important difference between ELM-FS and the other methods, i.e. LARS, MI-FW and MI-FWBW, is that the obtained FSP highlight features which must be selected together. Indeed, ELM-FS is able to drop a feature when the feature subset size increases,

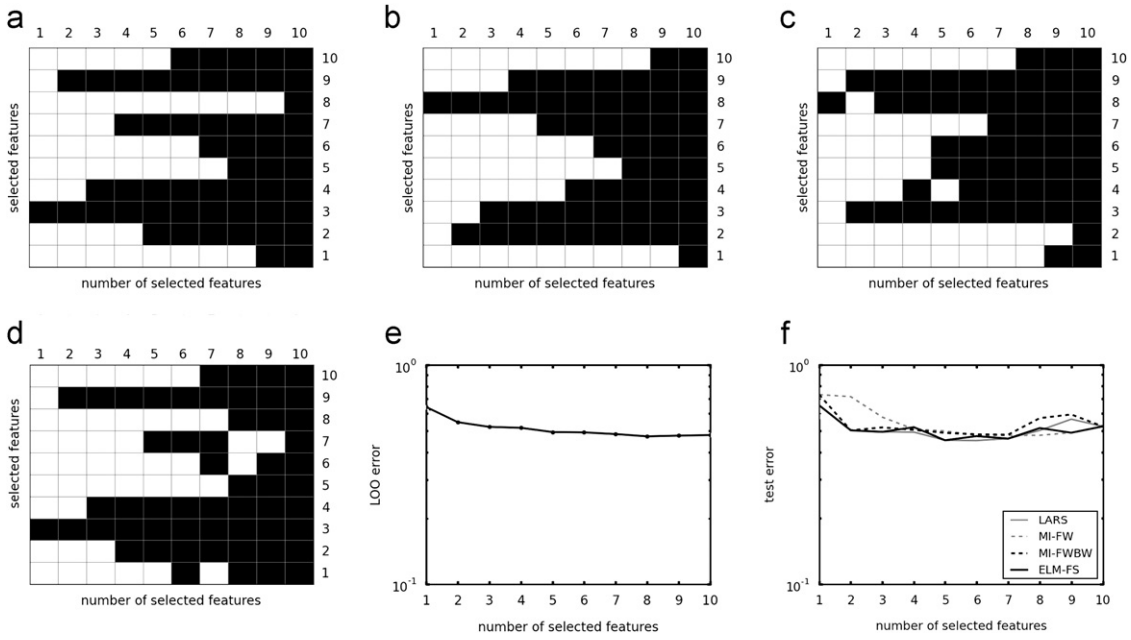

Fig. 7. Results for the diabetes dataset: (a-d) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (e) the SET curve for ELM-FS and (f) the test errors for the four compared methods. Notice the logarithmic scales for errors. (a) FSP: LARS. (b) FSP: MI-FW. (c) FSP: MI-FWBW. (d) FSP: ELM-FS. (e) SET curve: ELM-FS. (f) test errors.

in order to add two new features which must be used jointly. This provides an insightful information about the target function.

Table 1 shows the computation times for the different feature selection algorithms, including the computation time for the selection of the k parameter used by the Kraskov estimator of the mutual information. In terms of computation time, ELM-FS is comparable to MI-FWBW, whereas LARS and MI-FW are faster. However, it should be highlighted that (i) LARS only searches for linear relationships and (ii) MI-FW searches through a much smaller space of possible feature subsets.

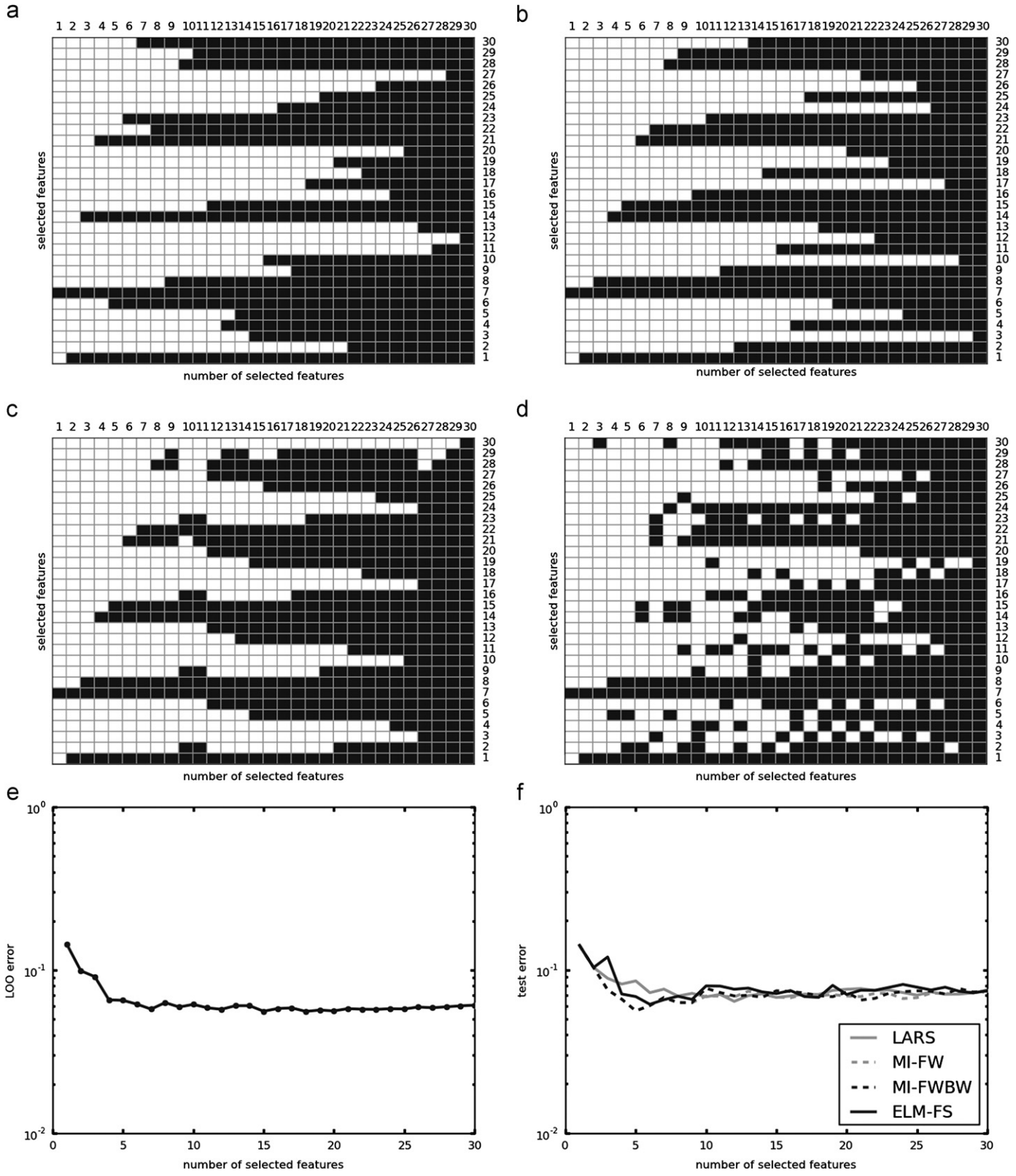

Fig. 8. Results for the Poland electricity load dataset: (a-d) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (e) the SET curve for ELM-FS and (f) the test errors for the four compared methods. Notice the logarithmic scales for errors.

5.3. Results on real datasets

In this subsection, four real datasets [25] are used to compare ELM-FS with LARS, MI-FW and MI-FWBW: (i) the diabetes dataset from Efron et al. [2], (ii) the Poland electricity load dataset [26], (iii) the Santa Fe laser dataset [27] and (iv) the anthrokids dataset [28]. The diabetes dataset consists of 442 samples with 10 continuous features. For comparison, a FSP is given for LARS in [2]. The Poland electricity load dataset consist of 1370 samples with 30 continuous features. The original time series is transformed into a regression problem, where the 30 past values are used to predict the electricity load of the next day. For example, the first feature corresponds to the last day. The Santa Fe laser dataset consists of 10,081 samples with 12 continuous features. The anthrokids dataset consists of 1019 samples with 53 features. For the experiments, the diabetes dataset, the Poland electricity load dataset and the anthrokids dataset are split into two parts: 70% of the instances are used for training and the remaining 30% of the instances are used for test. The Santa Fe laser dataset is split into a training set of 1000 instances and a test set of 9081 instances.

For the diabetes dataset, Fig. 7 shows (i) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (ii) the SET curve for ELM-FS and (iii) the test errors for the four methods. For feature subsets of at most three features, LARS and ELM-FS obtain lower test errors than MI-FW and MI-FWBW. The FSP of LARS and ELM-FS are identical for the three first subset sizes: feature 3 (body mass index), feature 9 (one of the serum measurements) and feature 4 (blood pressure). For larger feature subset sizes, the four algorithms achieve similar test errors. Here, ELM-FS has no advantage over other methods, but it achieves performances which are similar in terms of test error to those obtained by LARS, which it the best other method for this dataset. The SET curve provided by ELM-FS shows that using two or three features, almost optimal results can be achieved, which is confirmed by the test errors.

For the Poland electricity load dataset, Fig. 8 shows (i) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (ii) the SET curve for ELM-FS and (iii) the test errors for the four methods. According to the SET curve for ELM-FS, seven features are sufficient to achieve almost optimal generalisation error. For this subset size, LARS, MI-FW, MI-FWBW and ELM-FS choose the feature subsets {1,6,7,14,21, 23, 30} {1,7,8,14,15,21,22}, {1,7,8,14,15,21,22} and {1, 3,7, 8,21, 22,23}, respectively. In other words, the four methods recommend to use the electricity load of yesterday (feature 1) and the electricity load of previous weeks on the same day (e.g. features 7, 14 or 21). Moreover, they recommend to use the electricity load around these days (e.g. features 6, 8, 15 or 22), which could e.g. be used to estimate the time series derivative. A few other features are used (e.g. features 3, 23 and 30), which may be explained by the important amount of redundancy in this regression problem. Test errors are similar for the four methods.

For the Santa Fe laser dataset, Fig. 9 shows (i) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (ii) the SET curve for ELM-FS and (iii) the test errors for the four methods. For ELM-FS, the SET curve shows that 4 features are sufficient to achieve almost optimal results. The FSP for ELM-FS shows that the corresponding subset is {1, 2,4, 7}. But the FSP also shows that features 3 and 8 seem to be important. Here, the FSP provide additional information: the analysis of the successive feature subsets for smaller subset sizes reveals other interesting features. This cannot be seen if only the selected feature subset is considered. LARS, MI-FW and MI-FWBW do not select features 1, 2, 4, and 7 together for small feature subsets. It explains that ELM-FS beats them in terms of test error for these subsets sizes. Here, LARS needs eight features to achieves a similar test error, whereas the methods based on mutual information are not able to compare to ELM-FS. Notice that the FSP obtained using ELM-FS has many discontinuities, which suggests redundancy or complex interactions between the features and the target function.

For the anthrokids dataset, Figs. 10, 11 and 12 show (i) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (ii) the SET curve for ELM-FS and (iii) the test errors for the four methods. For ELM-FS, the SET curve shows that nine features are sufficient to achieve almost optimal results. The test error achieves its minimum around this point for all methods. No method seems to be significantly better than the others. Yet, the FSP for ELM-FS is different from the three other FSPs: whereas LARS, MI-FW and MI-FWBW choose successive feature subsets which are very similar by design, ELM-FS does not

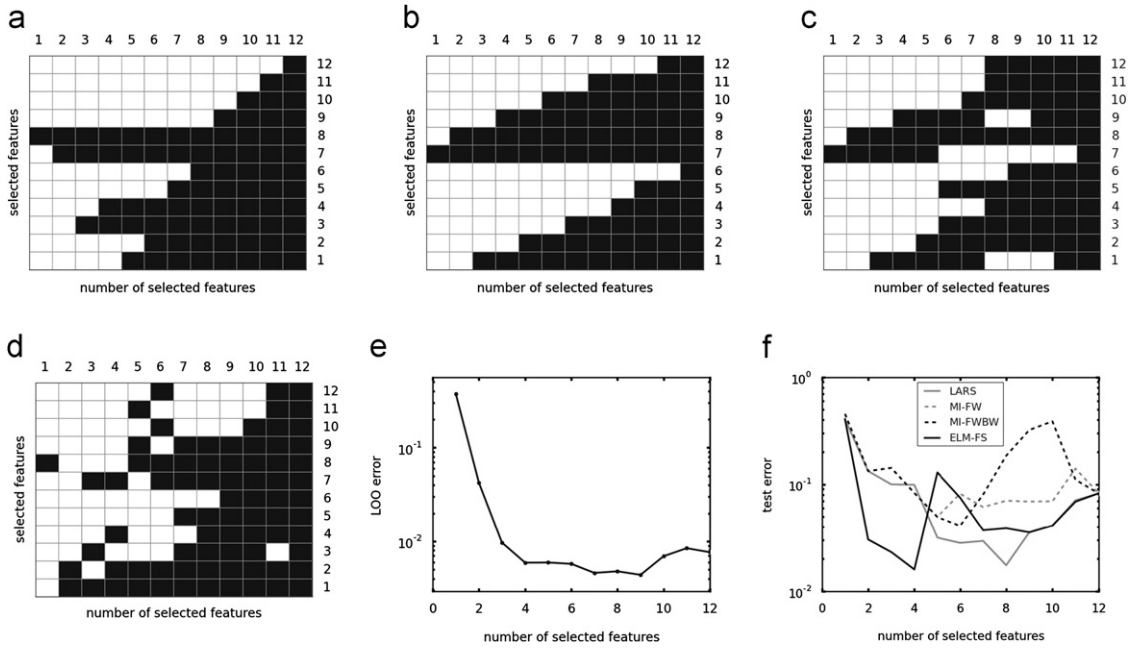

Fig. 9. Results for the Santa Fe laser dataset: (a-d) the FSPs for LARS, MI-FW, MI-FWBW and ELM-FS, (e) the SET curve for ELM-FS and (f) the test errors for the four compared methods. Notice the logarithmic scales for errors.

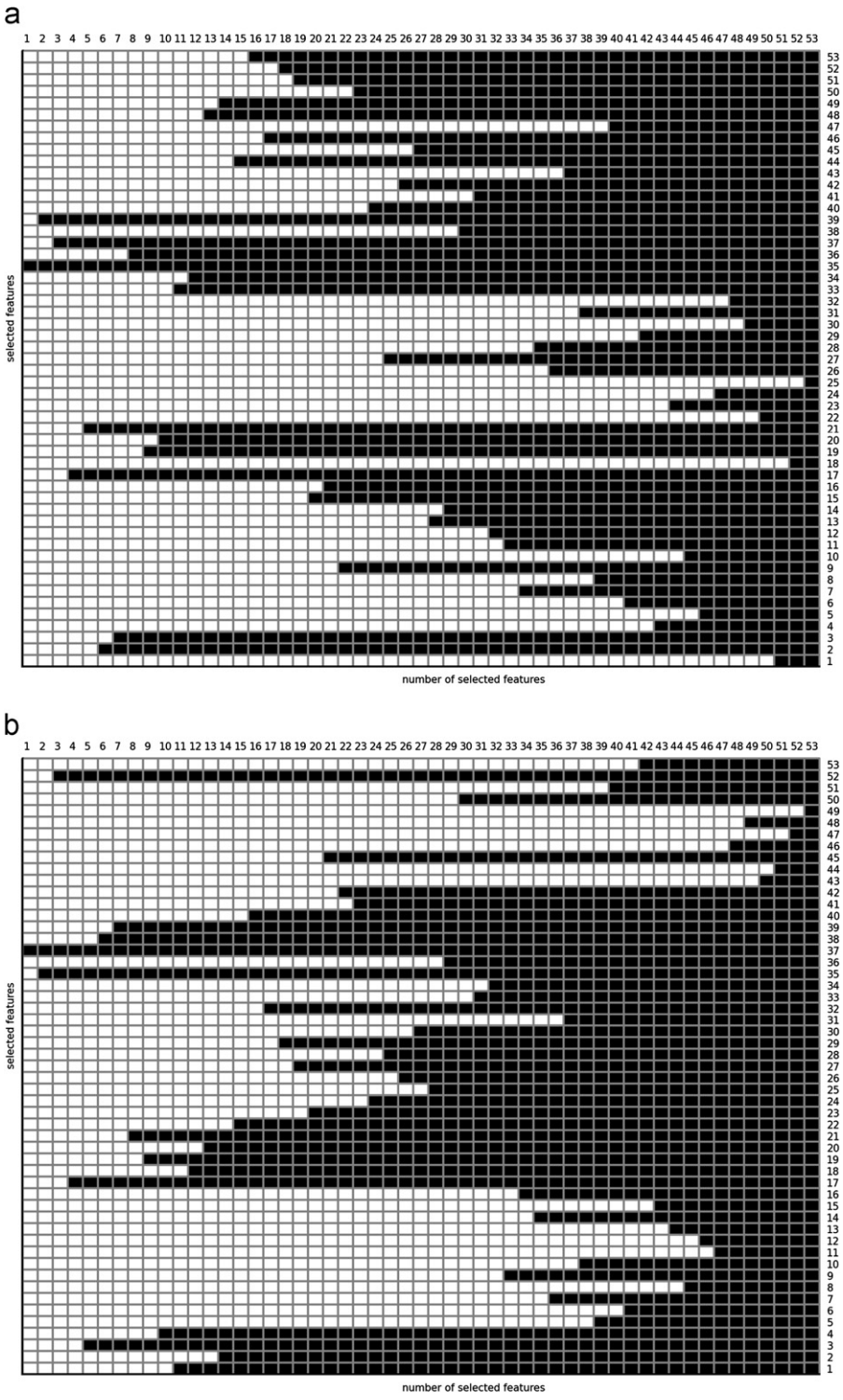

Fig. 10. Results for the anthrokids dataset: (a-b) the FSPs for LARS and MI-FW.

suffer from this constraint. The discontinuities in the FSP for ELM-FS indicate that there is an important amount of redundancy between features in this regression problem, what could not be seen with LARS, MI-FW and MI-FWBW. A closer analysis shows that three clusters of features are selected often in the nine first columns of the FSP for ELM-FS: features 1–3, 19–21 and 35–39. These three clusters are also found by the other feature selection methods. Notice that ELM-FS also selects e.g. features 8, 12 and 49 which are not selected by other methods.

Similarly to the case of artificial datasets, the results obtained in this subsection show that ELM-FS obtains sound feature subsets. For the diabetes dataset, the Poland electricity load dataset and the anthrokids dataset, ELM-FS is equivalent to the best methods in terms of test error. For all four datasets, the SET curve obtained by ELM-FS can be used to select the best feature subset size. For the Poland electricity load dataset and the Santa Fe laser dataset, the feature subset which corresponds to the minimum of the SET curve also obtains the minimum test error.

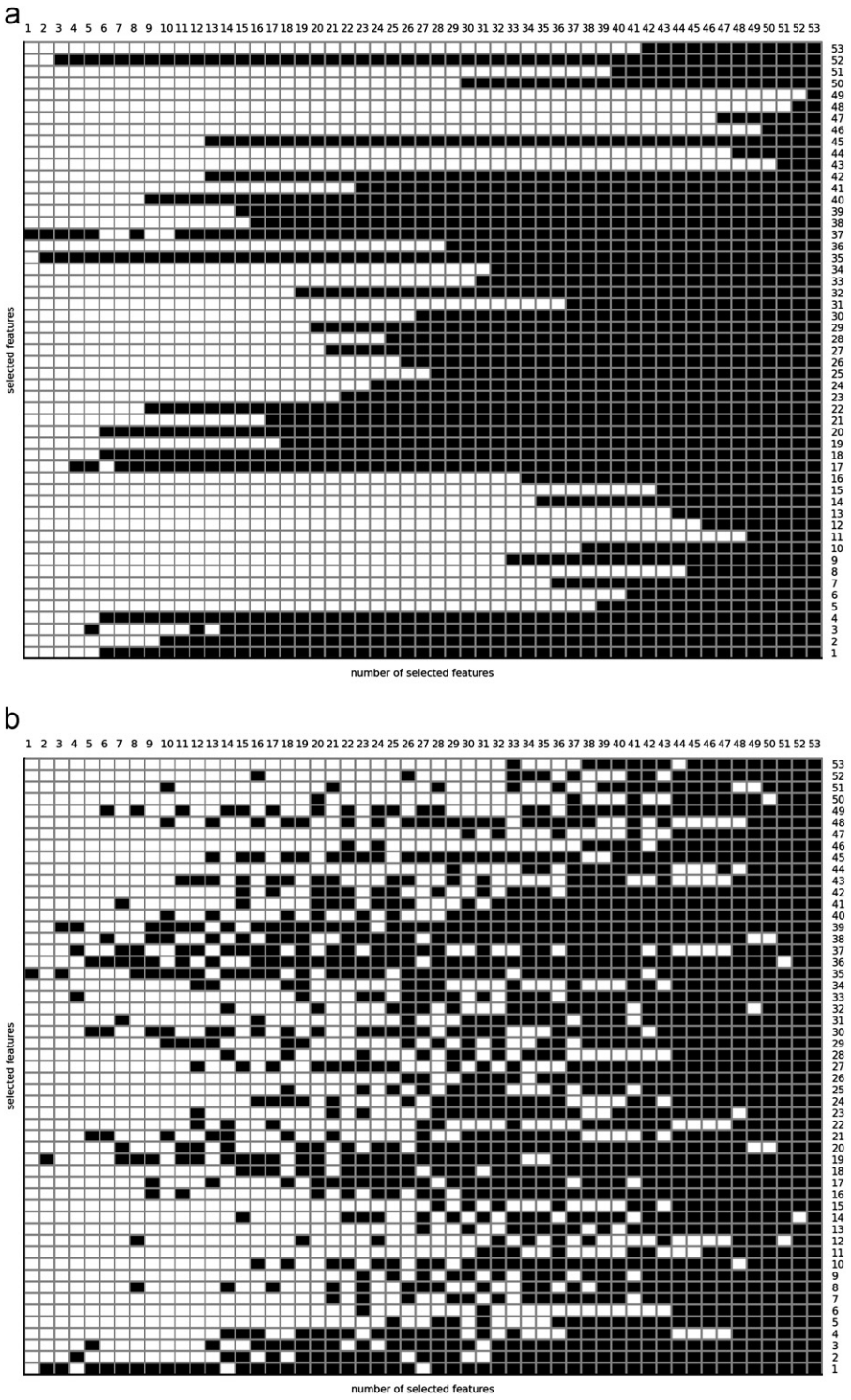

Fig. 11. Results for the anthrokids dataset: (a-b) the FSPs for MI-FWBW and ELM-FS.

For the diabetes dataset and the anthrokids dataset, the feature subset which corresponds to a sufficient LOO error in the SET curve almost obtains the minimum test error, with 3 and 9 features respectively.

The results for the Santa Fe laser dataset show that ELM-FS can be useful for problems with complex relationships between the features and the output. Firstly, the optimal test error is achieved with only four features, whereas LARS needs eight features to achieve a similar result. For the Santa Fe laser dataset, the small feature subsets obtained by ELM-FS allows reaching test errors which are significantly better (for the same subset sizes) than the test errors achieved by other methods. Secondly, the FSP obtained by ELM-FS reflects the complex relationships between the features and the target: there are many discontinuities in the FSP, which is also the case for the anthrokids dataset.

Table 2 shows the computation times for the different feature selection algorithms, including the computation time for the selection of the k parameter used by the Kraskov estimator of the mutual information. In terms of computation time, ELM-FS is comparable to MI-FWBW, whereas LARS and MI-FW are faster.

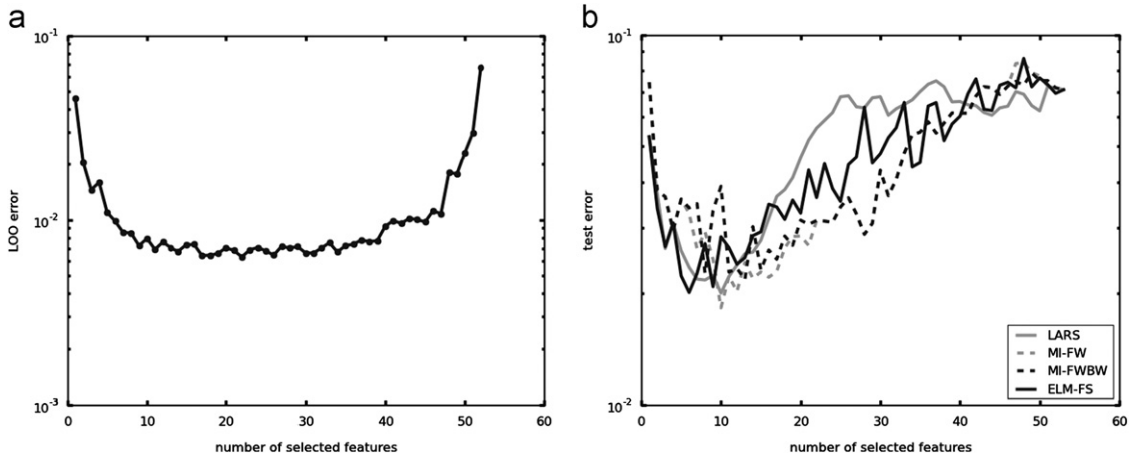

Fig. 12. Results for the anthrokids dataset: (a) the SET curve for ELM-FS and (b) the test errors for the four compared methods. Notice the logarithmic scales for errors.

Table 2
Computation times in seconds of the different feature selection algorithms for the diabetes dataset, the Poland electricity load dataset, the Santa Fe Laser dataset and the anthrokids dataset, including the search of the k parameter for the Kraskov estimator.

	LARS	MI-FW	MI-FWBW	ELM-FS
Diabetes	1.7e−3	1.2e+1	5.5e+1	6.0e+1
Poland electricity load	9.7e−3	4.9e+2	2.1e+3	7.1e+2
Santa Fe	2.9e−2	2.5e+2	7.0e+2	5.0e+2
Anthrokids	2.4e−2	4.1e+2	3.3e+3	4.5e+2

Again, it should be highlighted that (i) LARS only searches for linear relationships and (ii) MI-FW searches through a much smaller space of possible feature subsets.

6. Conclusion

This paper reviews two visual tools to help users and experts to perform feature selection and gain knowledge about the domain: the feature selection and the sparsity-error trade-off curve. The ELM-FS algorithm is proposed to build these two tools. A specific implementation using ELMs is used to analyse different datasets. The experimental results show that the proposed tools and the proposed algorithm can actually help users and experts. Indeed, they provide not only the optimal number of features but also the evolution of the estimation of the generalisation error, and which features are selected for different number of selected features. The proposed methodology allows making a trade-off between feature selection sparsity and generalisation error. This way, experts can e.g. reduce the number of features in order to design a model of the underlying process.

Acknowledgments

The authors would like to thank Gauthier Doquire (Université catholique de Louvain) who suggested the XOR problem for regression used in this paper.

References

[1] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction Springer Series in Statistics, 2nd ed., Springer, 2009.
[2] B. Efron, T. Hastie, L. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2004) 407–499.
[3] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E 69 (6) (2004) 066138, http://dx.doi.org/10.1103/PhysRevE.69.066138.
[4] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemometr. Intell. Lab. Syst. 80 (2) (2006) 215–226, http://dx.doi.org/10.1016/j.chemolab.2005.06.010.
[5] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[6] G.-B. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cyb. 2 (2011) 107–122.
[7] G.-B. Huang, D. Wang, Advances in extreme learning machines (elm2010), Neurocomputing 74 (16) (2011) 2411–2412.
[8] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cyb. B Cyb. 99 (2011) 1–17.
[9] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (16-18) (2008) 3460–3468, advances in Neural Information Processing (ICONIP 2006)/ Brazilian Symposium on Neural Networks (SBRN 2006).
[10] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally-pruned extreme learning machine, IEEE Trans. Neural Netw. 21 (1) (2010) 158–162, http://dx.doi.org/10.1109/TNN.2009.2036259.
[11] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th international joint conference on Artificial intelligence, vol. 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143.
[12] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Monographs on Statistics and Applied Probability, Chapman & Hall, 1993.
[13] D.M. Allen, The relationship between variable selection and data augmentation and a method for prediction, Technometrics 16 (1) (1974) 125–127.
[14] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. B 58 (1994) 267–288.
[15] J.M.F. Bach, R. Jenatton, G. Obozinski, Convex optimization with sparsity-inducing norms, in: S.J.W.S. Sra, S. Nowozin (Eds.), Optimization for Machine Learning, MIT Press, 2011.
[16] P. Bradley, O. L. Mangasarian, Feature selection via concave minimization and support vector machines, in: Machine Learning Proceedings of the Fifteenth International Conference (ICML 98), Morgan Kaufmann, 1998, pp. 82–90.
[17] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-norm support vector machines, in: S. Thrun, L.K. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems, vol. 16, MIT Press, 2004.
[18] E.K. Burke, G. Kendall (Eds.), Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer, 2006.
[19] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), 1st ed., Springer, 2007.
[20] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998.
[21] G.-B. Huang, C.-K. Siew, Extreme learning machine with randomly assigned RBF kernels, Int. J. Inf. Technol. 11 (1) (2005) 16–24.
[22] C. Rao, S. Mitra, Generalized Inverse of Matrices and its Applications, Wiley, 1971.
[23] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[24] M. Verleysen, F. Rossi, D. François, Advances in feature selection with mutual information, in: M. Biehl, B. Hammer, M. Verleysen, T. Villmann (Eds.), Similarity-Based Clustering, Lecture Notes in Computer Science, vol. 5400, Springer, Berlin/Heidelberg, 2009, pp. 52–69.
[25] Environmental and industrial machine learning group, http://research.ics.tkk.fi/eiml/datasets.shtml.

[26] A. Lendasse, J.A. Lee, V. Wertz, M. Verleysen, Forecasting electricity consumption using nonlinear projection and self-organizing maps, Neurocomputing 48 (1–4) (2002) 299–311.
[27] A.S. Weigend, N.A. Gershenfeld, Results of the time series prediction competition at the Santa Fe Institute, in: International Symposium on Neural Networks, 1993.
[28] A. Guillén, D. Sovilj, F. Mateo, I. Rojas, A. Lendasse, Minimizing the delta test for variable selection in regression problems, Int. J. High Perform. Syst. Archit. 1 (4) (2008) 269–281.

Benoît Frénay received an Engineer's degree from the Université catholique de Louvain (UCL), Belgium, in 2007. He is now a Ph.D. student at the UCL Machine Learning Group. His main research interests in machine learning include support vector machines, extreme learning, graphical models, classification, data clustering, probability density estimation and label noise.

Mark van Heeswijk has been working as an exchange student in both the EIML (Environmental and Industrial Machine Learning, previously TSPCi) Group and Computational Cognitive Systems Group on his Master's Thesis on "Adaptive Ensemble Models of Extreme Learning Machines for Time Series Prediction", which he completed in August 2009. Since September 2009, he started as a Ph.D. student in the EIML Group, ICS Department, Aalto University School of Science and Technology. His main research interest is in the field of high-performance computing and machine learning. In particular, how techniques and hardware from high-performance computing can be applied to meet the challenges one has to deal with in machine learning. He is also interested in biologically inspired computing, i.e. what can be learned from biology for use in machine learning algorithms and in turn what can be learned from simulations about biology. Some of his other related interests include: self-organization, complexity, emergence, evolution, bioinformatic processes, and multi-agent systems.

Yoan Miche was born in 1983 in France. He received an Engineer's Degree from the Institut National Polytechnique de Grenoble (INPG,France), and more specifically from TELECOM, INPG, on September 2006. He also graduated with a Master's Degree in Signal, Image and Telecom from ENSERG, INPG, at the same time. He recently received his Ph.D. degree in Computer Science and Signal and Image Processing from both the Aalto University School of Science and Technology (Finland) and the INPG (France). His main research interests are steganography/steganalysis and machine learning for classification/regression.

Michel Verleysen received the M.S. and Ph.D. degrees in electrical engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an invited professor at the Swiss E.P.F.L. (Ecole Polytechnique Fédérale de Lausanne, Switzerland) in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université ParisI-Panthéon-Sorbonne from 2002 to 2011, respectively. He is now a Full Professor at the Université catholique de Louvain, and Honorary Research Director of the Belgian F.N.R.S. (National Fund for Scientific Research). He is an editor-in-chief of the Neural Processing Letters journal (published by Springer), a chairman of the annual ESANN conference (European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning), a past associate editor of the IEEE Transactions on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He was the chairman of the IEEE Computational Intelligence Society Benelux chapter (2008–2010), and member of the executive board of the European Neural Networks Society (2005–2010). He is the author or co-author of more than 250 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?", in French, and of the "Nonlinear Dimensionality Reduction" book published by Springer in 2007. His research interests include machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.

Amaury Lendasse was born in 1972 in Belgium. He received the M.S. degree in Mechanical Engineering from the Universite Catholique de Louvain (Belgium) in 1996, M.S. in control in 1997 and Ph.D. in 2003 from the same university. In 2003, he has been a post-doctoral researcher in the Computational Neurodynamics Lab at the University of Memphis. Since 2004, he is a chief research scientist and a docent in the Adaptive Informatics Research Centre in the Aalto University School of Science and Technology (previously Helsinki University of Technology) in Finland. He has created and is leading the Environmental and Industrial Machine Learning (previously Time Series Prediction and Chemoinformatics) Group. He is chairman of the annual ESTSP conference (European Symposium on Time Series Prediction) and member of the editorial board and program committee of several journals and conferences on machine learning. He is the author or the coauthor of around 140 scientific papers in international journals, books or communications to conferences with reviewing committee. His research includes time series prediction, chemometrics, variable selection, noise variance estimation, determination of missing values in temporal databases, nonlinear approximation in financial problems, functional neural networks and classification.